

Amendments to the Claims

1. (CURRENTLY AMENDED) A microcontroller ~~(100)~~ the programming of which is carried out in at least one machine-dependent assembly language in which the assembler commands, with the exception of conditional program branches, are executable essentially independently of data,
_____in case of a fulfilled branch condition, for example, at least one fulfilled status flag, at least one program counter ~~(10)~~ being loadable with a new address and/or a new value, and
_____in case of an unfulfilled branch condition, for example, at least one unfulfilled status flag, the instruction being ended,
characterized in that in case of an unfulfilled branch condition the program counter ~~(10)~~, instead of ending the instruction, is optionally re-loadable with its previous address and/or with its previous value.
2. (CURRENTLY AMENDED) A microcontroller as claimed in claim 1, characterized by at least one multiplex unit ~~(20)~~ triggerable by means of the result of the testing of the branch condition, the input of the program counter ~~(10)~~
_____in case of a fulfilled branch condition being loadable with the new address and/or the new value, and
_____in case of an unfulfilled branch condition being loadable with the address at the output of the program counter ~~(10)~~ and/or with the value at the output of the program counter ~~(10)~~.
3. (CURRENTLY AMENDED) A microcontroller as claimed in ~~claim 1~~ or claim 1, characterized by its configuration as a smartcard controller.
4. (CURRENTLY AMENDED) An electrical or electronic device controlled by means of at least one microcontroller ~~(100)~~ according to ~~at least one of claims 1 to 3~~ claim 1.

5. (CURRENTLY AMENDED) A method for processing the programming of a microcontroller ~~(+00)~~ carried out in at least one machine-dependent assembly language,
_____the assembler commands, with the exception of conditional program branches, being executed essentially independently of data,
_____in case of a fulfilled branch condition, for example, at least one fulfilled status flag, at least one program counter ~~(+0)~~ being loaded with a new address and/or with a new value, and
_____in case of an unfulfilled branch condition, for example, at least one unfulfilled status flag, the instruction being ended,
characterized in that in case of an unfulfilled branch condition the program counter ~~(+0)~~, instead of ending the instruction, is optionally re-loaded with its previous address and/or with its previous value.

6. (CURRENTLY AMENDED) A method as claimed in claim 5, characterized in that
_____in case of a fulfilled branch condition the new address and/or the new value, and
_____in case of an unfulfilled branch condition the address at the output of the program counter ~~(+0)~~ and/or the value at the output of the program counter ~~(+0)~~ is/are supplied to the input of the program counter ~~(+0)~~.

7. (CURRENTLY AMENDED) A method as claimed in ~~claim 5 or 6~~ claim 5, characterized in that the testing of the branch condition and/or the loading of the program counter ~~(+0)~~ is/are carried out with complementary data.

8. (CURRENTLY AMENDED) A method as claimed in ~~at least one of claims 5 to 7~~ claim 5, characterized in that in case of an unfulfilled branch condition the option between ending the instruction and re-loading the program counter ~~(+0)~~ with its previous address and/or with its previous value is controlled by at least one special bit (so-called "select bit").

9. (CURRENTLY AMENDED) A method as claimed in ~~at least one of~~
~~claims 5 to 8~~claim 5, characterized in that, in case of an unfulfilled branch condition,
in program parts which are non-critical and/or not security-sensitive the option of
ending the instruction is selected in particular by the special bit.

10. (CURRENTLY AMENDED) A method as claimed in ~~claim 8 or 9~~claim 8,
characterized in that the special bit option can be switched on and off in any desired
sequence, for example, by means of at least one random function and/or by means of
at least one suitable bit sequence.